UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/729,448	12/04/2000	Joseph A. Porkka	MS150750.1	6259

27195   7590   08/02/2005

AMIN & TUROCY, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER
ROCHE, TRENTON J

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 08/02/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	09/729,448	PORKKA, JOSEPH A.
	Examiner	Art Unit	
	Trenton J. Roche	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>27 October 2004</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
   closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-35</u> is/are pending in the application.

   4a) Of the above claim(s) <u>5,6 and 26-35</u> is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-4 and 7-25</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>04 December 2000</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a)☐ All   b)☐ Some * c)☐ None of:

   1.☐ Certified copies of the priority documents have been received.

   2.☐ Certified copies of the priority documents have been received in Application No. _____.

   3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
      application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)   4)☐ Interview Summary (PTO-413)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)   Paper No(s)/Mail Date. _____ .
3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)   5)☐ Notice of Informal Patent Application (PTO-152)
    Paper No(s)/Mail Date _____.   6)☐ Other: _____.

DETAILED ACTION

Response to Arguments

1.   In view of the appeal brief filed on 27 October 2004, PROSECUTION IS HEREBY

REOPENED. New grounds of rejection are set forth below.

To avoid abandonment of the application, appellant must exercise one of the following two

options:

(1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37

CFR 1.113 (if this Office action is final); or,

(2) request reinstatement of the appeal.

If reinstatement of the appeal is requested, such request must be accompanied by a

supplemental appeal brief, but no new amendments, affidavits (37 CFR 1.130, 1.131 or 1.132) or

other evidence are permitted. See 37 CFR 1.193(b)(2).

2.   Claims 1-4 and 7-25 are currently pending.

3.   Claims 1-4 and 7-25 have been examined.

Per claims 1-4, 7-13 and 15-25:

Regarding claims 1-4 and 7-13, the Applicant states that Boehm et al., read according to the

alternative (single workstation) embodiment, does not disclose a plurality of build machines as

recited in independent claim 1, and that because there is only one build machine, the published files

are not shared by a plurality of build machines. In response to applicant's argument that the single

workstation embodiment does not disclose a plurality of build machines, it is noted that the claim

language states a first component *for* (emphasis added) building a list of file names of published files

to be shared (emphasis added) thereby implying intended use, and a recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim. In a claim drawn to a process of making, the intended use must result in a manipulative difference as compared to the prior art. See *In re Casey*, 370 F.2d 576, 152 USPQ 235 (CCPA 1967) and *In re Otto*, 312 F.2d 937, 939, 136 USPQ 458, 459 (CCPA 1963). Similarly, the claims further recite a second component *for* (emphasis added) distributing...*that are to be* (emphasis added) stored persistently, thereby implying intended use. Boehm et al. discloses a first component for building a list of file names, and a second component for distributing files, and as such, anticipates the claim language.

Regarding claims 15-18, the Applicant states that Boehm et al. does not disclose or suggest that published build files are broadcast to one or more build machines by other build machines. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., broadcasting to one or more build machines by other build machines) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). In the instance of claim 15, as the claim recites "one or more" language in relation to build machines and build files, then one file on one machine is sufficient to anticipate the claim. Because Boehm et al. can be operated as a single workstation, and it has been shown that the workstation has a component for broadcasting and a component for persistently storing, then anticipation of the claim is met.

Regarding claims 19-20, similarly to the arguments presented regarding claims 15-18, the Applicant states that Boehm et al. does not teach or suggest that build files are distributed to one or more build machines by other build machines or by a post-build machine. Again, the Applicant appears to be arguing limitations not recited in the claim language, as the claim only requires a single build file and a single build machine. Furthermore, given that only one file may be utilized in the system of claims 19 and 20, the step of "determining which of the one or more build files..." does not reasonably make sense given the situation that only one file is used. If the rest of the claim language is to be performed, then no real determination "which" of the files to be used would be made, but rather, only the one file would be used. As such, there is no real determination concerning which file to use; the system must use the only file provided.

Regarding claims 21-23, the Applicant states that Boehm et al. does not disclose or suggest the limitations recited. Concerning claim 21, a reasonable interpretation of the claim involves collecting one file from one machine, and "distributing" the file to the machine; effectively, accessing a file and storing the file again. As Boehm et al. is capable of being practiced on a single workstation, the claim is clearly anticipated. Furthermore, nothing in the claim would preclude the use of a network cache as in the multi-workstation embodiment of Boehm et al. Files collected from a build machine and placed on a network cache so that they can be further distributed to other build machines from the network cache would still anticipate the claim.

Regarding claim 24, the Applicant states that Boehm et al. does not teach or suggest collecting one or more build files from one or more build machines and for persistently storing the one or more files. In response, note the arguments presented above in relation to claims 19 and 20.

Similarly, the claim only requires a single file and a single build machine, of which Boehm et al.

discloses, and further, the step of "determining which of the transmitted build files to store

persistently" presents the same problematic situation as that described in connection with claims 19

and 20.

Regarding claim 25, note the arguments concerning claims 19 and 20.

For these reasons, the rejection regarding claims 1-4, 7-13 and 15-25 is proper and

maintained.

Per claim 14:

The Applicant states that the Examiner used hindsight in combining the teaching of Boehm

et al. with that of Lubkin et al. to come to the conclusion that the use of a hash as a data structure

would be obvious, and stated that such an assertion was based on advantages disclosed in the

applicant's specification. In response, it is noted that the use of a hash data structure for storing lists

and its associated advantages are grossly well known to those of ordinary skill in the art, however, as

no motivation is provided in the teachings of Lubkin et al. as to why a hash was utilized, a new

reference has been provided, and as such, the argument concerning hindsight is considered moot.

Election/Restrictions

4.   Applicant's election of Group I in Paper No. 4 is acknowledged. Because applicant did not

distinctly and specifically point out the supposed errors in the restriction requirement, the election

has been treated as an election without traverse (MPEP § 818.03(a)).

Claim Rejections - 35 USC § 112

5.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6.   Claims 19, 20 and 25 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for

failing to particularly point out and distinctly claim the subject matter which applicant regards as the

invention.

Independent claims 19, 20 and 25 allege to recite methods for building a software system,

however, the claim language appears to have numerous issues regarding the specifics of the build

system which renders the claim indefinite and essentially repetitive. An in-depth analysis shows that

each of the claims recite "one or more" language in relation to the build machines, files names, and

build files. As such, the claim is interpreted according to the broadest reasonable interpretation.

7.   Claims 19, 20 and 25 recites the limitation "the one or more build files" in lines 3, 4 and 3,

respectively. There is insufficient antecedent basis for this limitation in the claim.

It is indefinite as to whether the multiple references to "one or more build files" within the

claim language is referring to the same set of build files, or a different set of build files, or even to a

subset of the build files. As the claim language reads presently, it is unclear whether the files received

from the post build machine, intended to be stored on the build machine, are the same exact files

that had been initially sent to the post build machine, or is a reference to a different set or subset of

build files. For example, line 9 of independent claim 19 recites "sending the one or more build

files...," however, it is unclear whether this is in reference to the one or more build files collected in

the collecting step of the claim, or whether it is in reference to the one or more build files stored in

the persistently storing step of the claim, etc.

As such, the language recited in claims 19, 20 and 25 renders the claims highly indefinite, and

one of ordinary skill in the art would not be reasonable capable of determining the scope of the

claims.

Claim Rejections - 35 USC § 101

8.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any
> new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of
> this title.

9.   Claims 1 and 15 are rejected under 35 U.S.C. 101 because the claimed invention is directed

to non-statutory subject matter. A system for building a software system, the system comprising a

first component for building a list of published files, and a second component for distribution of the

published files is non-statutory, since it is not tangible embodied in a manner so as to be executable,

as the only execution is in an intended use statement. Rather, the claim currently only recites two

software components for performing some intended action, and as such, is directed to non-statutory

subject matter.

Claim Rejections - 35 USC § 102

10.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in
> the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent
> by another filed in the United States before the invention by the applicant for patent, except that an international

application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11.   Claims 1-4, 7-13 and 15-25 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S.

Patent 6,457,170 to Boehm et al.

Regarding claim 1:

Boehm et al teach:

- a first component for building a list of file names ("the present invention generates a cache

  link structure. The cache link structure is a system of source file links…" in col. 2 lines 46-

  48.)

- a second component for distributing to one or more of the build machines one or more

  published files, identified in the list of file names, that are to be stored persistently by the one

  or more build machines ("pre-loading one or more network cache memories with as much

  useful information as can be ascertained from the build list…" in col. 6 lines 2-4. Further,

  "the source file handler checks out a copy of the source file from the software library

  archive…renames the file…and copies the renamed file into the proper network cache…" in

  col. 9 lines 24-28. Further, as noted in col. 9 lines 10-14, "the present invention can be

  practiced with…one or more local caches, if the present invention is being practiced on a

  standalone workstation." The files are inherently stored persistently in the local cache of the

  standalone workstation.)

substantially as claimed.

Regarding claim 2:

The rejection of claim 1 is incorporated, and further, Boehm et al teach wherein the first

component, second component and the one or more build machines execute on a single computer

as claimed ("the present invention can be practiced with a single network cache that holds source

and object files..." in col. 9 lines 10-12)

Regarding claim 3:

The rejection of claim 1 is incorporated, and further, Boehm et al teach wherein the first

component, second component and the one or more build machines execute on a plurality of

computers as claimed ("In a preferred embodiment, there will be a plurality of network caches...one

for each different host architecture..." in col. 9 lines 6-9)

Regarding claim 4:

The rejection of claim 1 is incorporated, and further, Boehm et al teach wherein each the first

component, second component and the one or more build machines execute on separate computers

as claimed ("In a preferred embodiment, there will be a plurality of network caches...one for each

different host architecture..." in col. 9 lines 6-9)

Regarding claim 7:

The rejection of claim 1 is incorporated, and further, Boehm et al teach storing on a computer

readable medium ("a storage medium that includes a certain amount of quickly accessible electronic

storage..." in col. 5 lines 26-28)

Regarding claim 8:

The rejection of claim 1 is incorporated, and further, Boehm et al teach a list of published file names as claimed ("object file links are links from explicit object file names and potentially usable object files names to corresponding copies of the object file stored in cache." in col. 4 lines 16-19)

Regarding claim 9:

The rejection of claim 1 is incorporated, and further, Boehm et al teach initiating file transfers as claimed (Note Fig. 6 item 216 and the corresponding section of the disclosure. To copy to a network cache, a file transfer must inherently be initiated.)

Regarding claim 10

The rejection of claim 1 is incorporated, and further, Boehm et al teach initiating file transfers as claimed (Note Fig. 7A and 7B. To build the program, the builder must inherently initiate file transfers to the network caches for the purpose of receiving the information contained in the source files.)

Regarding claim 11:

The rejection of claim 1 is incorporated, and further, Boehm et al teach initiating file transfers as claimed ("Alternatively, software developers may want the option of manually updating the object file caches, by loading only selected object files…" in col. 15 lines 20-22. The file transfers are initiated by the user, which is a process other than those stated in the claim.)

Regarding claim 12:

The rejection of claim 1 is incorporated, and further, Boehm et al teach building a list of updates as claimed ("practitioners of the present invention may want to set up an automatic process for updating the object caches that initiates whenever a new build list is checked into RCS." in col. 16 lines 34-37)

Regarding claim 13:

The rejection of claim 1 is incorporated, and further, Boehm et al teach a data structure for storing the list of published file names (Note Fig. 8A, items 100 and 712. Data structures must inherently be used to store the information in these lists.)

Regarding claim 15:

Boehm et al teach:

- a component for broadcasting to one or more of the build machines one or more published build files ("pre-loading one or more network cache memories with as much useful information as can be ascertained from the build list…" in col. 6 lines 2-4. Further, "the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28)

- a component for selectively receiving and storing persistently one or more of the broadcast published build files ("the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28. For the file to be stored in the network cache, a component must have received and stored the file. Further, as noted in col. 9 lines 10-14,

"the present invention can be practiced with...one or more local caches, if the present

invention is being practiced on a standalone workstation." The files are inherently stored

persistently in the local cache of the standalone workstation.)

substantially as claimed.

Regarding claim 16:

The rejection of claim 15 is incorporated, and further, Boehm et al teach a component for

broadcasting as claimed (Note rejection regarding claim 2. The processes of the invention are shown

to operate on one computer.)

Regarding claim 17:

The rejection of claim 15 is incorporated, and further, Boehm et al teach a component for

broadcasting as claimed (Note rejection regarding claim 3. The processes of the invention are shown

to operate on a plurality of computers.)

Regarding claim 18:

The rejection of claim 15 is incorporated, and further, Boehm et al teach a component for

broadcasting as claimed (Note rejection regarding claim 4. The processes of the invention are shown

to operate on separate computers.)

Regarding claim 19:

Boehm et al teach:

- a method for building a software system ("a method and apparatus for building a software

    system..." in col. 2 lines 35-36)

- collecting from one or more build machines, one or more files names corresponding to the

    one or more build files ("the present invention generates a cache link structure. The cache

    link structure is a system of source file links..." in col. 2 lines 46-48.)

- determining which of the one or more build files the one or more build machines are to send

    to a post build machine (Note Fig. 4, item 400 and the corresponding section of the

    disclosure. Cache updating is performed via a post-build 'machine,' which is controlled by

    the central network controller.)

- persistently storing the one or more build files on the one or more build machines ("the

    source file handler checks out a copy of the source file from the software library

    archive...renames the file...and copies the renamed file into the proper network cache..." in

    col. 9 lines 24-28. Further, as noted in col. 9 lines 10-14, "the present invention can be

    practiced with...one or more local caches, if the present invention is being practiced on a

    standalone workstation." The files are inherently stored persistently in the local cache of the

    standalone workstation.)

- sending to the one or more build machines a list of file names of build files to be sent to the

    post build machine, and sending the one or more build files to the post build machine

    ("practitioners of the present invention may want to set up an automatic process for

    updating the object caches that initiates whenever a new build list is checked into RCS." in

    col. 16 lines 34-37. If the build object is found to need updating, it would be sent to the post

    build machine for updating.)

- determining which of the one or more build files the one or more build machines are to receive from the post build machine (Note Fig. 4, item 400 and the corresponding section of the disclosure. The system would inherently determine which of the updated build files the network caches should be updated with.)

- sending to the one or more build machines a list of file names of build files to be received from the post build machine ("the present invention generates a cache link structure. The cache link structure is a system of source file links…" in col. 2 lines 46-48.)

- distributing one or more build files to the one or more build machines ("pre-loading one or more network cache memories with as much useful information as can be ascertained from the build list…" in col. 6 lines 2-4. Further, "the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28)

- persistently storing the one or more build files distributed to the one or more build machines ("the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28. For the file to be stored in the network cache, a component must have stored the file. Further, as noted in col. 9 lines 10-14, "the present invention can be practiced with…one or more local caches, if the present invention is being practiced on a standalone workstation." The files are inherently stored persistently in the local cache of the standalone workstation.)

substantially as claimed.

Regarding claim 20:

Boehm et al teach a computer-readable medium having computer-executable instructions. Note

rejection regarding claim 7. Further, this claim is directed to a computer readable medium having

stored thereon computer-executable instructions for performing the method of claim 19, and is

therefore rejected for the reasons set forth in connection with claim 19.

Regarding claim 21:

Boehm et al teach:

- a method for building a software system ("a method and apparatus for building a software

  system..." in col. 2 lines 35-36)

- collecting one or more build files from one or more build machines (Note Fig. 7A and 7B.

  To build the program, the builder must inherently collect files from the network caches for

  the purpose of receiving the information contained in the source files.)

- distributing the one or more build files to the one or more build machines ("the source file

  handler checks out a copy of the source file from the software library archive...renames the

  file...and copies the renamed file into the proper network cache..." in col. 9 lines 24-28)

Regarding claim 22:

The rejection of claim 21 is incorporated, and further, Boehm et al teach broadcasting the one or

more build files to the one or more build machines ("the source file handler checks out a copy of

the source file from the software library archive...renames the file...and copies the renamed file into

the proper network cache..." in col. 9 lines 24-28)

Regarding claim 23:

The rejection of claim 22 is incorporated, and further, Boehm et al teach determining and storing the files as claimed ("pre-loading one or more network cache memories with as much useful information as can be ascertained from the build list…" in col. 6 lines 2-4. Further, "the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28)

Regarding claim 24:

Boehm et al teach:

- a computer-readable medium having stored thereon computer-executable instructions ("a storage medium that includes a certain amount of quickly accessible electronic storage…" in col. 5 lines 26-28)

- collecting one or more build files from one or more build machines (Note Fig. 7A and 7B. To build the program, the builder must inherently collect files from the network caches for the purpose of receiving the information contained in the source files.)

- distributing the one or more build files to the one or more build machines ("the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28)

- determining which of the transmitted build files to store persistently, and persistently storing one or more of the transmitted files ("pre-loading one or more network cache memories with as much useful information as can be ascertained from the build list…" in col. 6 lines 2-4. Further, "the source file handler checks out a copy of the source file from the software library archive…renames the file…and copies the renamed file into the proper network cache…" in col. 9 lines 24-28. Finally, as noted in col. 9 lines 10-14, "the present invention

can be practiced with...one or more local caches, if the present invention is being practiced

on a standalone workstation." The files are inherently stored persistently in the local cache of

the standalone workstation.)

Regarding claim 25:

Boehm et al teach:

- a system for building a software system ("a method and apparatus for building a software

  system..." in col. 2 lines 35-36)

- means for collecting from one or more build machines, one or more files names

  corresponding to the one or more build files ("the present invention generates a cache link

  structure. The cache link structure is a system of source file links..." in col. 2 lines 46-48.)

- means for determining which of the one or more build files the one or more build machines

  are to send to a post build machine (Note Fig. 4, item 400 and the corresponding section of

  the disclosure. Cache updating is performed via a post-build 'machine,' which is controlled

  by the central network controller.)

- means for sending the one or more build files to the post build machine ("practitioners of

  the present invention may want to set up an automatic process for updating the object

  caches that initiates whenever a new build list is checked into RCS." in col. 16 lines 34-37. If

  the build object is found to need updating, it would be sent to the post build machine for

  updating.)

- means for determining which of the one or more build files the one or more build machines

  are to receive from the post build machine (Note Fig. 4, item 400 and the corresponding

section of the disclosure. The system would inherently determine which of the updated build

files the network caches should be updated with.)

- means for sending to the one or more build machines a list of file names of build files to be

    sent to the one or more build machines and the build files to be received from the one or

    more build machine ("the present invention generates a cache link structure. The cache link

    structure is a system of source file links..." in col. 2 lines 46-48. Further, Note Fig. 8A, items

    100 and 712 which indicate which files to be received from the build machines for the

    purpose of building the complete program.)

- means for distributing one or more build files to the one or more build machines ("pre-

    loading one or more network cache memories with as much useful information as can be

    ascertained from the build list..." in col. 6 lines 2-4. Further, "the source file handler checks

    out a copy of the source file from the software library archive...renames the file...and

    copies the renamed file into the proper network cache..." in col. 9 lines 24-28)

- means for persistently storing one or more build files distributed to the one or more build

    machines persistently one or more of the broadcast published build files ("the source file

    handler checks out a copy of the source file from the software library archive...renames the

    file...and copies the renamed file into the proper network cache..." in col. 9 lines 24-28. For

    the file to be stored in the network cache, a component must have received and stored the

    file. Further, as noted in col. 9 lines 10-14, "the present invention can be practiced

    with...one or more local caches, if the present invention is being practiced on a standalone

    workstation." The files are inherently stored persistently in the local cache of the standalone

    workstation.)

Claim Rejections - 35 USC § 103

12.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness

rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section
> 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the
> subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary
> skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the
> invention was made.

13.   Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,457,170

to Boehm et al in view of Microsoft Press Computer Dictionary, Second Edition, hereafter referred

to as Microsoft.

Regarding claim 14:

The rejection of claim 13 is incorporated, and further, Boehm et al. do not teach a data structure

utilizing a hash as claimed. Microsoft teaches that in a system utilizing lists it is well known to utilize

a hash to search the list ("Hash searches are highly efficient because the hashing enables direct (or

perhaps almost direct) access to the target element." on page 193.) It would have been obvious to

one of ordinary skill in the art at the time the invention was made to utilize a hash with the list of

published file names as disclosed by Boehm et al., as this would enable a user to efficiently search

the list and be capable of direct or almost direct accessing of target elements in the list disclosed by
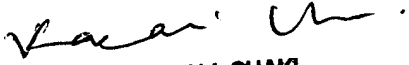
Boehm et al.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner

should be directed to Trenton J. Roche whose telephone number is (571) 272-3733. The examiner

can normally be reached on Monday - Friday, 9:00 am - 5:30 pm.

14.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor,

Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where

this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system,

see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system,

contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Trenton J Roche
Examiner
Art Unit 2193

TJR

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100